

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА УПРАВЛЕНИЯ МЕДИКО-БИОЛОГИЧЕСКИМИ СИСТЕМАМИ

Харламов Кирилл Александрович

Выпускная квалификационная работа бакалавра

**Создание программного обеспечения для анализа
динамики курсов валют, построения и
прогнозирования валютной корзины минимальной
волатильности**

Направление 010300

Фундаментальные информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Корников В. В.

Санкт-Петербург

2016

Содержание

| | |
|--|----|
| Введение | 3 |
| Постановка задачи | 5 |
| Обзор литературы | 6 |
| Глава 1. Экономическая модель | 8 |
| 1.1. Модель обмена | 8 |
| 1.2. Инвариантные индексы | 10 |
| 1.3. Валютные корзины | 12 |
| Глава 2. Инструменты программной реализации | 14 |
| 2.1. Язык программирования | 14 |
| 2.2. Графический интерфейс | 16 |
| 2.3. Построение графиков | 17 |
| Глава 3. Детали программной реализации | 18 |
| 3.1. Входные данные | 18 |
| 3.2. Подсчет индексов | 20 |
| 3.3. Построение валютных корзин | 22 |
| 3.4. Использование генетического алгоритма при построении корзины минимальной волатильности | 24 |
| 3.5. Построение прогноза на основе линейной регрессии | 29 |
| Глава 4. Инструкция по использованию программы | 31 |
| 4.1. Стартовый экран настроек | 31 |
| 4.2. Визуализация индексов валют и валютных корзин | 35 |
| 4.3. Прогнозирование | 38 |
| Заключение | 41 |
| Список литературы | 43 |

Введение

Валютой называют важнейшую составляющую денежной системы какого-либо государства или региона, являющуюся мерой стоимости, средством обращения и платежа. Только Центральный банк Российской Федерации ведет статистику по курсам 149 валют [1], однако также существуют виртуальные валюты, валюты непризнанных или частично признанных государств и территорий, частные и региональные валюты. Численные значения курсов валют в совокупности достаточно объемны и мало показательны. В связи с этим возникает необходимость создания простого и гибкого инструмента для обработки данных подобного рода.

Валютный рынок, как и любой другой, подвержен колебаниям и изменениям, то есть является нестабильным. Во многом это происходит из-за того, что цены формируются на основе спроса и предложения, которые, в свою очередь, постоянными не являются. Это влечет за собой малоустойчивость, а зачастую и нелогичность в формировании цен на отдельные валюты. Следовательно, имеет смысл использовать наборы валют, взятых в определенных пропорциях – валютные корзины, которые глубже отражают состояние рынка и его динамику. Более того, располагая статистическими данными за некоторый период времени и используя ряд математических методов, возможно построить максимально стабильную корзину.

Пусть ценность некоторого блага выражается в определенной валюте. Имея построенную стабильную валютную корзину, можно получить курс обмена интересующей нас валюты к данной корзине. Таким образом получим показатель ценности блага в стабильной агрегированной денежной единице. Это позволит наблюдать за реальной динамикой ценности актива, которая не отражает в себе колебания конкретной валюты.

Другим применением стабильных валютных корзин являются международные торговые операции по форвардным контрактам. В таких контрактах часто дата оплаты поставки какого-либо товара не совпадает с датой самой поставки. Данный факт вносит в бизнес определенные риски, связанные с возможными изменениями курсов валют за промежуток времени между передачей товара и получением за него оплаты. Использование стабильной валютной корзины вместо одной или нескольких конкретных валют при подобных торговых операциях позволило бы избежать проблем.

Кроме того, многие государства хранят резервы в иностранных валютах. При падении на валютном рынке используемых государством валют, денежные запасы страны уменьшатся. Таким образом, стабильные валютные корзины также позволяют эффективно сохранять денежный капитал.

Наконец, имея статистические данные, можно построить прогноз, показывающий предположительные значения и динамику курсов отдельных валют и валютных корзин. Прогноз по очевидным причинам не может быть абсолютно точным. Но, не смотря на это, грамотное построение прогноза является важным аспектом экономической деятельности.

В виду широкого распространения информационных технологий и высокой вычислительной мощности современных персональных компьютеров, удобным инструментом для выполнения вышеописанных действий может являться специальное программное обеспечение. Программная реализация позволяет скрыть от конечного пользователя математические расчеты и выполнить обработку значительных объемов информации за весьма короткий промежуток времени.

Постановка задачи

В данной работе ставится задача рассмотреть экономическую модель валютного рынка, включающую в себя инвариантные индексы и валютные корзины, а также создать специализированное программное обеспечение (ПО). Это программное обеспечение должно позволять проводить анализ динамики курсов валют, строить валютные корзины (в том числе максимально стабильные), делать прогноз изменений стоимости валют и валютных корзин. Разрабатываемая программа имеет исследовательскую направленность, однако вместе с тем должна быть достаточно простой в освоении. Важным аспектом является предоставление пользователю возможности использовать входные статистические данные, получаемые из общедоступных источников, что придает программному обеспечению необходимую гибкость и универсальность. Описываемое ПО должно являться удобным инструментом для исследователя, позволяющим экспериментировать с минимальными затратами времени на расчеты и изменение параметров модели.

Обзор литературы

Простая модель обмена, на основе которой строится модель предметной области в данной работе, вводится в совместных публикациях научных сотрудников СПбГУ: кандидата физико-математических наук, доцента кафедры высшей математики факультета прикладной математики – процессов управления – Геннадия Михайловича Хитрова, и профессора экономического факультета – Николая Васильевича Хованова [2, 3, 4].

Проблема построения агрегированных валют минимального риска также широко освещается в публикациях Николая Васильевича Хованова. В его работах обосновывается необходимость рассмотрения составных валют, строятся модели обмена и индексы простых и составных благ, а также описываются методы построения составных благ минимальной волатильности [5, 6, 7].

В свою очередь, опираясь на результаты Хованова и его соавторов, другие исследователи создали ряд работ по смежным темам. Например, Victor Pontines и Ramkishen S. Rajan в исследовательской статье «The Asian Currency Unit (ACU): exploring alternative currency weights» рассматривают возможность применения альтернативных весовых коэффициентов для Азиатской валютной единицы [8].

Поскольку для выполнения поставленной в данной работе задачи необходимо использование ряда технических средств и алгоритмов, выделим литературу и из этой области. В свете использования в созданной программе генетического алгоритма минимизации, стоит отметить известную книгу Джона Холланда «Адаптация в естественных и искусственных системах» [9], которая стала толчком для популяризации и дальнейших исследований в области эволюционных алгоритмов. В наши дни существует множество публикаций и книг, в которых можно найти информацию о генетических алгоритмах. Например, принципы построения таких алгоритмов изложены в

книге Юрия Ивановича Нечаева и Александра Борисовича Дегтярева «Интеллектуальные системы: концепция и приложения» [10].

В качестве пособия по использованному в данной работе языку программирования С# автором использовалась книга «Язык программирования С#» Андерса Хейлсберга – одного из разработчиков этого языка [11].

Глава 1. Экономическая модель

1.1. Модель обмена

При разработке программного обеспечения, прежде всего, необходимо формализовать предметную область, к которой оно относится. В данной работе предметной областью является валютный рынок (его часть), следовательно, можно построить следующую модель.

Рассмотрим набор экономических благ $G = \{g_1, \dots, g_n\}$, которые в нашем случае будут являться валютами. Каждая из валют может иметь количественную характеристику – действительное число $q_i \geq 0$, $i = 1, \dots, n$, $q_1 + \dots + q_n > 0$.

Любая валюта может быть обменяна на определенное количество другой валюты. Таким образом, коэффициент обмена для двух валют g_i и g_j определим как

$$c(i, j) = \frac{q_j}{q_i}.$$

Для полноты функционала программного обеспечения необходимо иметь значения коэффициентов $c(i, j)$ для всех рассматриваемых валют. Набор коэффициентов $c(i, j)$ составляет матрицу обмена $C = \{c(i, j)\}$, $i = 1, \dots, n$, $j = 1, \dots, n$. Эта информация по большей части будет относиться к априорной, получаемой из внешнего источника, и называться курсами валют. Отношение обмена, которое отражают коэффициенты, является отношением эквивалентности, так как для него характерны рефлексивность, симметричность и транзитивность:

- любую валюту можно обменивать саму на себя;
- если валюту a можно обменивать на валюту b , то и валюту b можно обменивать на валюту a ;

- если валюту a можно обменять на валюту b , а валюту b на валюту c , то валюту a можно обменять на валюту c .

Свойство транзитивности играет особенно важную роль при обработке входных данных в программе, так как из внешнего источника не всегда возможно получить все значения матрицы $C = \{c(i, j)\}$, ввиду неполноты исходных данных или из соображений компактности их хранения. При выполнении данного свойства для расчета всех значений матрицы обмена достаточно знать одну её строку.

Для того чтобы модель была полностью адекватной (то есть соответствовала предметной области), необходимо отметить два свойства матрицы C . Первое, наиболее очевидное: все диагональные элементы матрицы обмена равны единице, так как одна единица некоторой валюты обменивается на одну единицу той же самой валюты. Второе: если $c(i, j)$ единиц валюты i обменивается на одну единицу валюты j , то одна единица валюты i обменивается на $1/c(i, j)$ единиц валюты j .

1.2. Инвариантные индексы

Элементы матрицы обмена $C = \{c(i, j)\}$ отражают динамику валюты i по отношению к валюте j и наоборот. По изменению лишь одного курса $c(i, j)$ трудно понять, какой из переменных оно вызвано. Поэтому для отслеживания реальной ситуации на валютном рынке встает вопрос о введении характеристики, которая отражала бы динамику некоторого курса по отношению не к одной, а ко всем имеющимся в рассмотрении валютам. Именно зависимость от всех имеющихся валют означает, что такой индекс будет независим от базового блага, то есть инвариантен по отношению к каждой конкретной денежной единице. Это позволит наиболее объективным образом оценивать изменения в ценности валют. Можно составить бесконечно много различных индексов. В данной работе используются аддитивные и мультипликативные индексы следующего вида:

$$IndAdd(i) = \frac{c(i, j)}{\frac{1}{n} \sum_{i=1}^n c(i, j)}, \text{ где } j \in \{1, \dots, n\};$$
$$IndMult(i) = \frac{c(i, j)}{\sqrt[n]{\prod_{i=1}^n c(i, j)}}, \text{ где } j \in \{1, \dots, n\}.$$

Значение данных индексов для валюты i не зависит от выбора базовой валюты j . Далее необходимо ввести в рассмотрение последовательность моментов времени $t \in \{1, \dots, T\}$, в которые измеряются значения курсов для входных данных. Соответственно, инвариантные индексы будут зависеть от t :

$$IndAdd(i, t) = \frac{c(i, j, t)}{\frac{1}{n} \sum_{i=1}^n c(i, j, t)}, \text{ где } j \in \{1, \dots, n\}, t \in \{1, \dots, T\};$$
$$IndMult(i, t) = \frac{c(i, j, t)}{\sqrt[n]{\prod_{i=1}^n c(i, j, t)}}, \text{ где } j \in \{1, \dots, n\}, t \in \{1, \dots, T\}.$$

Полученные индексы могут иметь сильно различающиеся числовые значения, что осложняет восприятие визуализированной на графике

динамики изменения валюты. Поэтому из соображений удобства будет разумно редуцировать индексы следующим образом:

$$\begin{aligned} IndAddR\left(i, \frac{t}{t_0}\right) &= \frac{IndAdd(i, t)}{IndAdd(i, t_0)}; \\ IndMultR\left(i, \frac{t}{t_0}\right) &= \frac{IndMult(i, t)}{IndMult(i, t_0)}. \end{aligned}$$

Здесь t_0 — некоторый момент времени. В программе в качестве t_0 используется начальный момент времени.

1.3. Валютные корзины

Валютная корзина – набор определенных количеств различных валют [12]. Необходимость в использовании валютной корзины возникает в связи с нестабильностью отдельных валют. Именно совокупность нескольких валют, взятых в определенных пропорциях, позволяет комплексно учитывать рыночную ситуацию и изменение общеэкономических условий обмена. Количество той или иной валюты, входящей в рассматриваемую корзину, называется удельным весом валюты. Каждый удельный вес принадлежит отрезку $[0; 1]$, а сумма всех весов должна равняться единице.

Валютные корзины, как и отдельные валюты, по сути, являются расчётными единицами. Поэтому из тех же соображений, что и для валют, имеет смысл для формализации понятия «валютная корзина» использовать редуцированный индекс, не зависящий от базовых валют, выбранных для построения индексов компонентов корзины. Искомая величина будет являться функцией от инвариантных индексов отдельных валют, приведенных к начальному моменту времени:

$$CBind\left(w, \frac{t}{t_0}\right) = \sum_{i=1}^n w_i IndR\left(i, \frac{t}{t_0}\right),$$

где n – число валют в корзине. Здесь и далее под $IndR\left(i, \frac{t}{t_0}\right)$ будем подразумевать аддитивный или мультипликативный индекс (применимы оба). Ограничения на значения удельных весов $w = (w_1, \dots, w_n)$ имеют вид:

$$\sum_{i=1}^n w_i = 1, \quad w_i \geq 0 \quad \forall i \in \{1, \dots, n\}. \quad (1.3.1)$$

Аддитивный вид индекса позволяет работать с корзинами, некоторые компоненты которых имеют близкий к нулю вес.

Далее перейдем к проблеме построения стабильной валютной корзины. Значение индекса корзины минимальной волатильности с течением времени должно изменяться минимальным образом (оставаться неизменным в

идеальном случае). Нестабильность корзины можно выразить через дисперсию:

$$D = \frac{1}{T} \sum_{t=1}^T (CBind(w, t) - MCBind(w))^2, \quad (1.3.2)$$

$$MCBind(w) = \frac{1}{T} \sum_{t=1}^T CBind(w, t),$$

$\{1, \dots, T\}$ – рассматриваемые моменты времени.

Таким образом, для построения валютной корзины минимальной волатильности необходимо подобрать весовые коэффициенты w_i таким образом, чтобы дисперсия D была наименьшей. Этого можно достичь, применив алгоритм минимизации к (1.3.2) при ограничениях (1.3.1). В данной работе использовался генетический алгоритм, который описан в параграфе 3.4.

Глава 2. Инструменты программной реализации

В данной главе описан набор инструментов и технологий, с помощью которых строится конечный программный продукт.

2.1. Язык программирования

Для создания программы выбран язык программирования C#. Он был разработан в 2002 году компанией Microsoft и с тех пор является одним из самых распространенных языков разработки. C# предназначен для программной платформы .NET Framework, которая, в свою очередь, ориентирована на операционную систему Microsoft Windows (хотя и существуют некоторые технологии, позволяющие работать с .NET Framework на других операционных системах). С учетом того, что по разным оценкам более 85% персональных ПК работает под управлением Windows [13, 14], написанная на C# программа может быть использована достаточно широким кругом пользователей, что удовлетворяет поставленным целям.

Данный язык имеет большое количество библиотек, в том числе и стандартных, что делает удобным процесс разработки и позволяет в некоторых модулях программы использовать уже готовые решения.

Так как C# имеет управляемый код, то есть исполнение кода осуществляется под управлением виртуальной машины .NET, пиковая производительность кода в нем не достижима (в отличие от неуправляемого кода). Однако этого и не требует концепция разработки программы в текущей работе, так как целевой аудиторией разработанного программного обеспечения являются частные лица, а не коммерческие организации. Следовательно, ранее описанные преимущества перевешивают тот факт, что предельная скорость выполнения программы может быть не максимальной.

Таким образом, язык C# является подходящим инструментом для решения поставленной в данной работе задачи.

2.2. Графический интерфейс

Графический интерфейс пользователя, ГИП (graphical user interface, GUI) – разновидность реализации пользовательского интерфейса, при которой использование программы осуществляется путем взаимодействия с графическими элементами, размещенными на экране.

Альтернативой GUI является интерфейс на основе командной строки, который является неприемлемым в данной работе. Аргументы в пользу реализации ГИП следующие:

- необходимость в максимальной «дружественности» интерфейса;
- существование большого объема информации, которую нужно визуализировать в различных формах;

Иными словами, трудозатраты и объем потраченного времени на разработку графического интерфейса в данном случае вполне оправданы.

Для реализации GUI был выбран интерфейс программирования приложений Windows Forms, который является частью .NET Framework. Для создания программы, удовлетворяющей требованиям, описанным в этой работе, требуется ряд широко используемых графических элементов:

- кнопки
- текстовые поля
- списки
- таблицы
- элементы для построения изображений

Все они присутствуют в Windows Forms, кроме того, могут быть созданы с использованием визуального редактора, что экономит время на разработку.

При этом ввиду исследовательского характера приложения не требуются какие-либо оригинальные и нестандартные дизайнерские решения.

Следовательно, данный интерфейс программирования вполне пригоден для использования.

2.3. Построение графиков

В ходе работы программы в виде входных необработанных данных, либо в результате обработки этих данных, возникает большой объем информации. В качестве этой информации выступают курсы валют, индексы валют, индексы различных валютных корзин, а также спрогнозированные значения временных рядов. Разработанное программное обеспечение позволяет работать со статистическими данными за десятки лет, объем этих данных после обработки и построения индексов значительно возрастает. Например, при использовании информации о курсах пяти разных валют к рублю за один год, лишь после расчёта полной матрицы обмена и инвариантных индексов получится 7200 числовых значений. Всю эту информацию необходимо представить пользователю наглядно. Так как существуют две основные характеристики: момент времени и значение величины, то лучшим выходом является построение двумерных графиков.

Язык C# в купе с Windows Forms позволяют разработать модуль программы, ответственный за создание 2D графиков. Однако в открытом доступе существует множество готовых библиотек, реализующих поставленную задачу с большим количеством инструментов и широкими возможностями настройки. Одной из таких библиотек является ZedGraph, которая имеет открытый код, полноценную документацию, хорошее быстродействие и стабильность в работе [15]. Именно она и была использована в данном проекте.

Глава 3. Детали программной реализации

3.1. Входные данные

Входными данными служат файлы формата DBF. Данный формат широко распространен. Его используют в базах данных для хранения информации предметной области и метаданных. Файлы этого формата с курсами различных валют к рублю можно получить на официальном сайте Центрального банка Российской Федерации [1]. Для просмотра содержимого файла можно воспользоваться, например, Microsoft Excel. ЦБ РФ предоставляет данные с 1992 года до наших дней. Существует возможность выбрать любую часть данного временного промежутка.

В программе используются два столбца из входных файлов: дата, когда было зафиксировано определенное значение курса, и само это значение. Получение этих данных из файла осуществляется при помощи SQL запроса, с применением набора интерфейсов OLE DB. Информация из каждого файла с курсом некоторой валюты к рублю извлекается в экземпляры класса `System.Data.DataTable`, представляющие таблицы. Затем эти таблицы объединяются в `System.Data.DataSet`.

Далее данные с курсами валют реорганизуются в набор матриц, где каждая матрица является матрицей обмена, введенной в параграфе 1.1, в некоторый момент времени (в данном случае – за какую-то дату). Неизвестные элементы матриц рассчитываются из исходных данных за счет того, что для курсов валют выполняется свойство транзитивности. В новом виде информация по-прежнему представлена экземпляром класса `DataSet`, состоящим из таблиц `DataTable`, только теперь каждая таблица относится не к конкретной валюте, а к моменту времени. Также создается массив дат, размер которого равен числу таблиц в новом `DataSet`.

Для наглядности вышеописанные преобразования для четырех валют (включая рубль) показаны на рисунке 1.3.1.

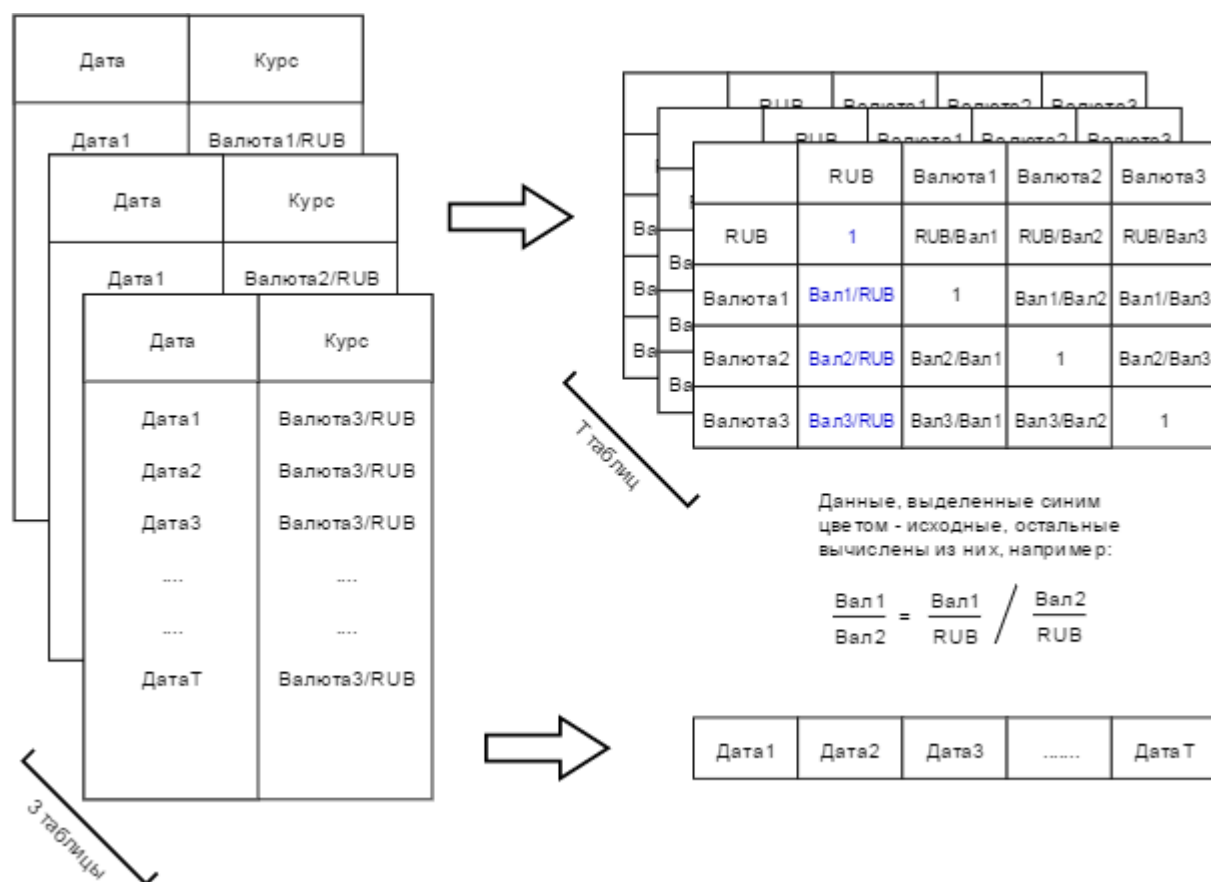


Рисунок 1.3.1. Слева: набор таблиц без преобразований; справа: полученный после реорганизации данных набор матриц, а также вектор дат, указывающих, когда проводились измерения курсов.

Полученные матрицы данных выводятся на экран в виде таблиц с применением элемента Windows Forms – DataGridView, в котором в качестве источника данных указывается одна из таблиц DataTable (матрица курсов). Номер выбранной таблицы определяется номером в массиве дат выбранной в календаре даты. Календарь также реализован средствами Windows Forms (элемент MonthCalendar). Помимо этого, на основе курсов валют строится 2D график при помощи библиотеки ZedGraph.

3.2. Подсчет индексов

После успешного получения входных данных и их обработки происходит подсчет инвариантных индексов, которые являются объективным показателем динамики курса той или иной валюты, используются для построения валютных корзин и при прогнозировании.

Согласно параграфу 1.2, в данной работе используются аддитивные индексы вида

$$Ind(i, t) = \frac{c(i, j, t)}{\frac{1}{n} \sum_{i=1}^n c(i, j, t)}, \quad (3.2.1)$$

либо мультипликативные

$$Ind(i, t) = \frac{c(i, j, t)}{\sqrt[n]{\prod_{i=1}^n c(i, j, t)}}, \quad (3.2.2)$$

$$j \in \{1, \dots, n\}, t \in \{1, \dots, T\},$$

которые приводятся к начальному моменту времени:

$$IndR\left(i, \frac{t}{t_0}\right) = \frac{Ind(i, t)}{Ind(i, t_0)}. \quad (3.2.3)$$

Выбор между двумя вышеприведенными типами осуществляет пользователь.

На языке C# функция, отвечающая за подсчет аддитивных индексов, выглядит следующим образом:

```
double[,] CreateAdditiveIndices(int baseCurrency) // baseCurrency - номер базовой валюты
{
    double[,] index = new double[numberOfCurrency, numberOfTimeMoments];
    double indexT0 = 1; // Индекс в начальный момент времени
    for (j = 0; j < numberOfCurrency; j++) // j - номер валюты
        for (t = 0; t < numberOfTimeMoments; t++) // t - момент времени
        {
            double sum = 0;
            for (int i = 0; i < numberOfCurrency; i++) // подсчет суммы в (3.2.1)
                sum = sum + rates[t][i][baseCurrency];
            index[j, t] = rates[t][j][baseCurrency] / (sum / numberOfCurrency);
            if (t == 0)
                indexT0 = index[j, 0];
            index[j, t] = index[j, t] / indexT0; // (3.2.3)
        }
    return index;
}
```

Для мультипликативных:

```
double[,] CreateMultiplicativeIndices(int baseCurrency) // baseCurrency - номер базовой
валюты
{
    double[,] index = new double[numberOfCurrency, numberOfTimeMoments];
    double indexT0 = 1; // Индекс в начальный момент времени
    for (j = 0; j < numberOfCurrency; j++) // j - номер валюты
        for (t = 0; t < numberOfTimeMoments; t++) // t - момент времени
        {
            double comp = 1;
            for (int i = 0; i < numberOfCurrency; i++) // подсчет произведения в
(3.2.2)
                comp = comp * rates[t][i][baseCurrency];
            index[j, t] = Pow(comp, 1.0 / numberOfCurrency);
            if (t == 0)
                indexT0 = index[j, 0]; // Индекс в начальный момент времени
            index[j, t] = index[j, t] / indexT0; // (3.2.3)
        }
    return index;
}
```

Из соображений краткости и наглядности в приведенном коде экземпляры класса DataSet заменены на многомерные массивы.

По окончании подсчетов, данные полученного массива по аналогии с курсами валют выводятся на экран в качестве таблицы, а также могут быть визуализированы в виде графика для любой валюты, выбранной пользователем (из имеющихся в рассмотрении).

3.3. Построение валютных корзин

Как отмечалось в параграфе 1.3, валютные корзины представляются в виде индексов, подсчитанных по формуле:

$$CBind\left(w, \frac{t}{t_0}\right) = \sum_{i=1}^n w_i IndR\left(i, \frac{t}{t_0}\right), \quad (3.3.1)$$

Построение корзины становится доступно только после подсчета индексов отдельных валют $IndR\left(i, \frac{t}{t_0}\right)$. Создается список с названиями имеющихся валют, в котором выбираются нужные для построения корзины. По названиям валют можно определить их номера в массиве индексов. Весовые коэффициенты w_i вводятся пользователем в текстовое поле через пробел самостоятельно.

Далее необходимо произвести парсинг весовых коэффициентов. Во-первых, необходимо предусмотреть некорректный ввод данных в текстовое поле (например, наличие букв среди чисел). Ошибочный текст исправляется путем удаления несоответствующих требованиям символов, приводится к единому формату. Во-вторых, могут быть не соблюдены ограничения, накладываемые на весовые коэффициенты рассматриваемой моделью (сумма весов равна единице, каждый вес принадлежит отрезку $[0; 1]$, число введенных весов должно совпадать с количеством выбранных валют в списке). Сообщение об ошибке такого рода выводится на экран.

Обработка содержимого текстового поля производится с помощью регулярных выражений. Например, убрать некорректные символы, можно следующим образом:

```
string patternWrongSymb = @"[^0123456789,. ]";  
string replacementWS = "";  
string text = Regex.Replace(input, patternWrongSymb, replacementWS);
```

В приведенном примере любой символ, не являющийся цифрой, точкой, запятой или пробелом, заменяется пустой строкой.

Если после парсинга весовые коэффициенты признаются корректными, происходит подсчет значений индекса валютной корзины и вывод их на график.

3.4. Использование генетического алгоритма при построении корзины минимальной волатильности

Согласно параграфу 1.3, задача построения корзины минимальной волатильности сводится к минимизации относительно $w = (w_1, \dots, w_n)$ функции

$$D = \frac{1}{T} \sum_{t=1}^T (CBind(w, t) - MCBind(w))^2, \quad (3.4.1)$$

$$MCBind(w) = \frac{1}{T} \sum_{t=1}^T CBind(w, t),$$

$\{1, \dots, T\}$ – рассматриваемые моменты времени,

при ограничениях

$$\sum_{i=1}^n w_i = 1, \quad w_i \geq 0 \quad \forall i \in \{1, \dots, n\}. \quad (3.4.2)$$

В данной работе для поиска оптимальных весовых коэффициентов использовался генетический алгоритм минимизации.

Генетический алгоритм (ГА) оптимизации – это алгоритм поиска оптимальных решений задачи путем эвристического подбора и модификаций искомых параметров с применением механизмов, аналогичных механизмам естественного отбора в природе. Термин «генетический алгоритм» впервые был упомянут Р. Багли в 1957 году, позднее, в конце 60-х годов двадцатого века, формальный механизм и теоретическое обоснование алгоритма было введено американским ученым Дж. Холландом.

Генетические алгоритмы имеют ряд достоинств [10, 16, 17, 18]:

- ГА не накладывают никаких требований на вид оптимизируемой функции и на вид ограничений. При применении ГА не появляется необходимость упрощать исследуемую модель в ущерб её адекватности.

- Генетические алгоритмы устойчивы к попаданию в локальный минимум, благодаря тому, что рассматривается множество альтернативных решений, которые к тому же подвергаются стохастическим изменениям.
- ГА удобен и достаточно прост в программной реализации, не требует использования методов численного дифференцирования.

Основным недостатком является скорость поиска экстремума. Однако в виду исследовательской направленности целевой программы и достаточной вычислительной мощности современных ПК, данный недостаток не является критичным.

Существует множество различных вариаций рассматриваемого алгоритма, ниже приводится описание конкретной реализации, использованной в данной работе.

Исходной задачей является задача поиска набора параметров $w = (w_1, \dots, w_n)$, при котором функция (3.4.1) примет минимальное значение, с учетом ограничений (3.4.2). Введем основные понятия. Fitness-функция – это функция, оценивающая меру пригодности некоторого потенциального решения. В данном случае fitness-функцией является выражение (3.4.1): чем меньше его значение для рассматриваемых параметров, тем ближе данный набор к окончательному результату. Популяцией будем называть набор из n векторов w . Особь – один из элементов популяции.

В начале работы алгоритма случайным образом задаются вектора популяции. Выбор наиболее подходящего размера популяции осуществляется опытным путем. После генерации значений каждого вектора к нему применяется функция нормализации, которая поддерживает ограничения (3.4.2). Также для каждого вектора вычисляется значение функции полезности. Это значение хранится в одной структуре данных с самим вектором. Таким образом, мы получили стартовую популяцию. Её элементы являются далекими от оптимальных. Теперь над особями из

популяции нужно произвести итеративную последовательность преобразований, чтобы увеличить их «полезность». Одна итерация данных преобразований называется шагом эволюции или эпохой. Оптимальное количество шагов эволюции, также как и размер популяции, определяется опытным путем.

Рассмотрим один эволюционный шаг. Он состоит из следующих составляющих.

1. Кроссовер. Случайным образом выбираются две особи популяции, называемые родительскими. Путем «скрещивания», то есть создания комбинаций параметров родителей, создаются две новых особи. В данной реализации алгоритма описанный процесс происходит по формулам:

$$r = \text{Random}(0; 1)$$

$$\alpha = 2r - 0.5$$

$$\beta = 2(1 - r) - 0.5$$

$$child1_i = |\alpha \cdot para_i + \beta \cdot mama_i|$$

$$child2_i = |\alpha \cdot mama_i + \beta \cdot para_i|$$

Эти формулы итерационно применяются для всех элементов вектора коэффициентов w . Здесь r – случайное число от 0 до 1. Ранее выбранные родительские особи обозначены, как $mama$ и $para$, а дочерние, соответственно, $child1$ и $child2$.

К новым особям применяется функция нормализации для выполнения наложенных на них ограничений, после чего к каждой новой особи прикрепляется значение $fitness$ -функции, подсчитанное подстановкой вектора параметров в функцию (3.4.1). Далее эти элементы добавляются в общую популяцию.

2. Мутация. На данном этапе случайным образом выбираются одна из особей популяции и один из её параметров. В этот параметр вносятся изменения:

$$w(i) = w(i) * \text{Random}(0; 1) + 0.5$$

Затем вектор коэффициентов с одной обновленной составляющей нормализуется и для него пересчитывается значение функции пригодности.

3. Редукция. В ходе выполнения текущей итерации популяция выросла на два элемента. Однако целесообразно поддерживать стабильную численность. Для этого отсортируем элементы популяции по значению fitness-функции и удалим два наименее подходящих элемента. В нашем случае таковыми будут те векторы весовых коэффициентов, при которых значения минимизируемой функции будут наибольшими.

Описанные три пункта повторяются в каждой эпохе эволюции. По окончании эволюционного процесса, получим набор отсортированных по оптимальности векторов, с помощью которых можно построить валютную корзину минимальной волатильности.

Блок-схема описанного алгоритма представлена на рисунке 3.4.1.

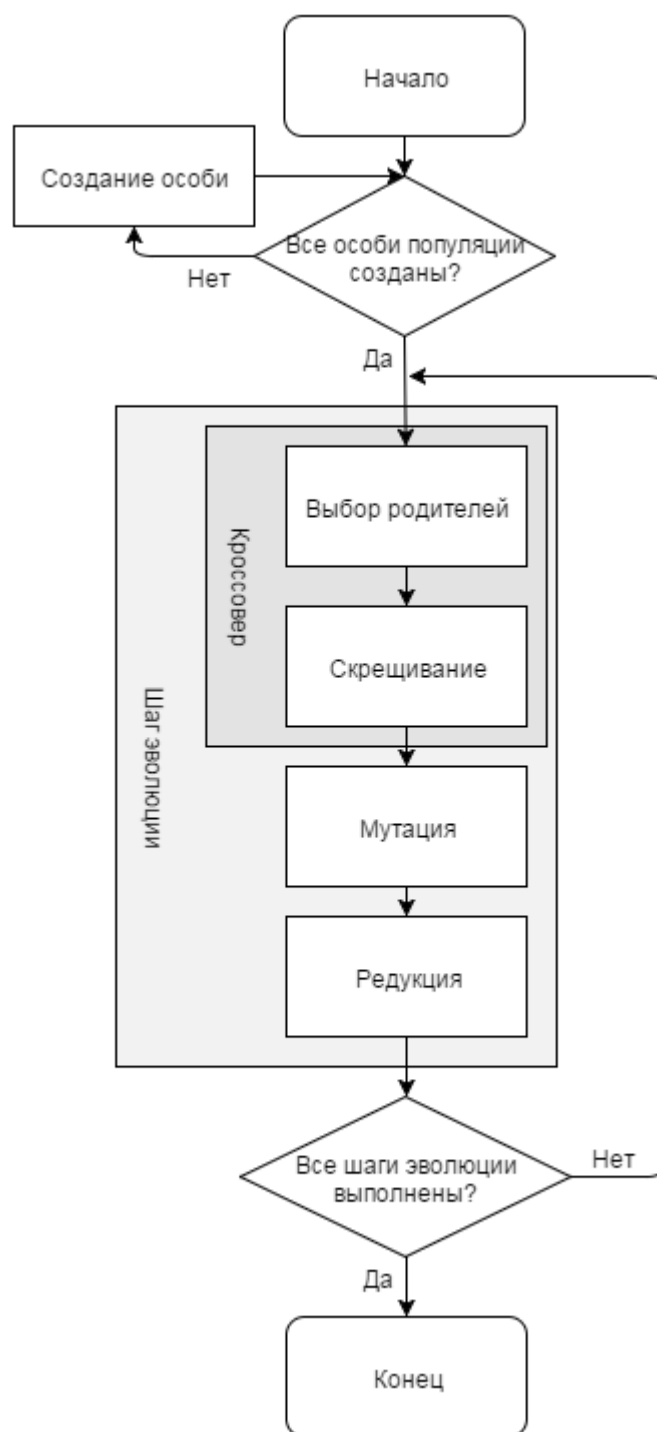


Рисунок 3.4.1. Блок-схема генетического алгоритма.

3.5. Построение прогноза на основе линейной регрессии

В качестве дополнительного инструмента в созданной программе реализован механизм построения прогноза курсов валют, индексов валют и валютных корзин. Рассмотрим применение для этих целей многофакторной линейной регрессии.

Целью построения регрессионной модели является выявление зависимости переменной Y от k независимых переменных X , называемых факторами. Пусть y – наперед известное значение, а Y – значение функции регрессии (спрогнозированное значение), тогда выполняется:

$$y = Y(a, X) + \varepsilon,$$

где ε – ошибка регрессионной модели. Функция регрессии задается в виде:

$$Y(a, X) = a_1 X_1 + \dots + a_k X_k. \quad (3.5.1)$$

X_i – факторы регрессии, a_i – коэффициенты регрессии.

Сформулируем задачу прогнозирования, применительно к рассматриваемому программному обеспечению. После обработки входных данных мы имеем значения курсов и индексов валют, а также индексов валютных корзин за заданный входными данными временной промежуток. Этот промежуток представлен моментами времени $\{1, \dots, T\}$. Далее для краткости будем называть любые значения (курсов или индексов) временным рядом. Необходимо построить значения временного ряда для моментов времени $\{T + 1, \dots, L\}$, где L – длина прогноза.

Перед тем, как начать использовать модель регрессии для непосредственного построения прогноза, необходимо определить значения коэффициентов регрессии. Для этого найдем такую линейную функцию Y , которая наилучшим образом аппроксимирует имеющий временной ряд (или его часть). Коэффициенты данной функции и будут искомыми значениями $a = \{a_1, \dots, a_k\}$. Для решения поставленной промежуточной задачи определим меру качества аппроксимации:

$$E = \sum_{i=1}^N (Y_{app}(i) - Y_{real}(i))^2,$$

где $N \leq T$ – число аппроксимируемых моментов времени, Y_{app} – функция аппроксимации, Y_{real} – реальные значения временного ряда. Необходимо минимизировать E по коэффициентам $\{a_1, \dots, a_k\}$. Для этого в программе применяется описанный в предыдущем параграфе генетический алгоритм. Пользователю предоставляется возможность изменить заданные по умолчанию размер популяции и количество эпох эволюции через соответствующие текстовые поля. Также пользователю необходимо указать количество независимых переменных в процентном соотношении от общего количества моментов времени.

Теперь, зная вектор $a = \{a_1, \dots, a_k\}$, можно приступить к прогнозированию. Получив из текстового поля длину прогноза L , по формуле (3.5.1) вычисляется значение временного ряда на момент времени $T + 1$. Далее по той же формуле считается значение на момент $T + 2$. При этом в качестве одной из независимых переменных уже используется подсчитанное на предыдущем шаге значение за дату $T + 1$. Процесс продолжается до достижения $T + L$. При этом на каждом шаге общее количество независимых переменных остается постоянным. Таким образом, точность прогноза будет убывать с удалением от стартовой точки.

Доверительный интервал прогноза построим следующим образом. Для каждой прогнозной точки *forecast* найдем верхнюю и нижнюю границы интервала a и b соответственно:

$$a = forecast + t \cdot s;$$

$$b = forecast - t \cdot s.$$

Здесь t – коэффициент Стьюдента для доверительной вероятности 0.95, s – стандартное отклонение значений временного ряда.

Глава 4. Инструкция по использованию программы

4.1. Стартовый экран настроек

После запуска программы открывается стартовое окно (рисунок 4.1.1).

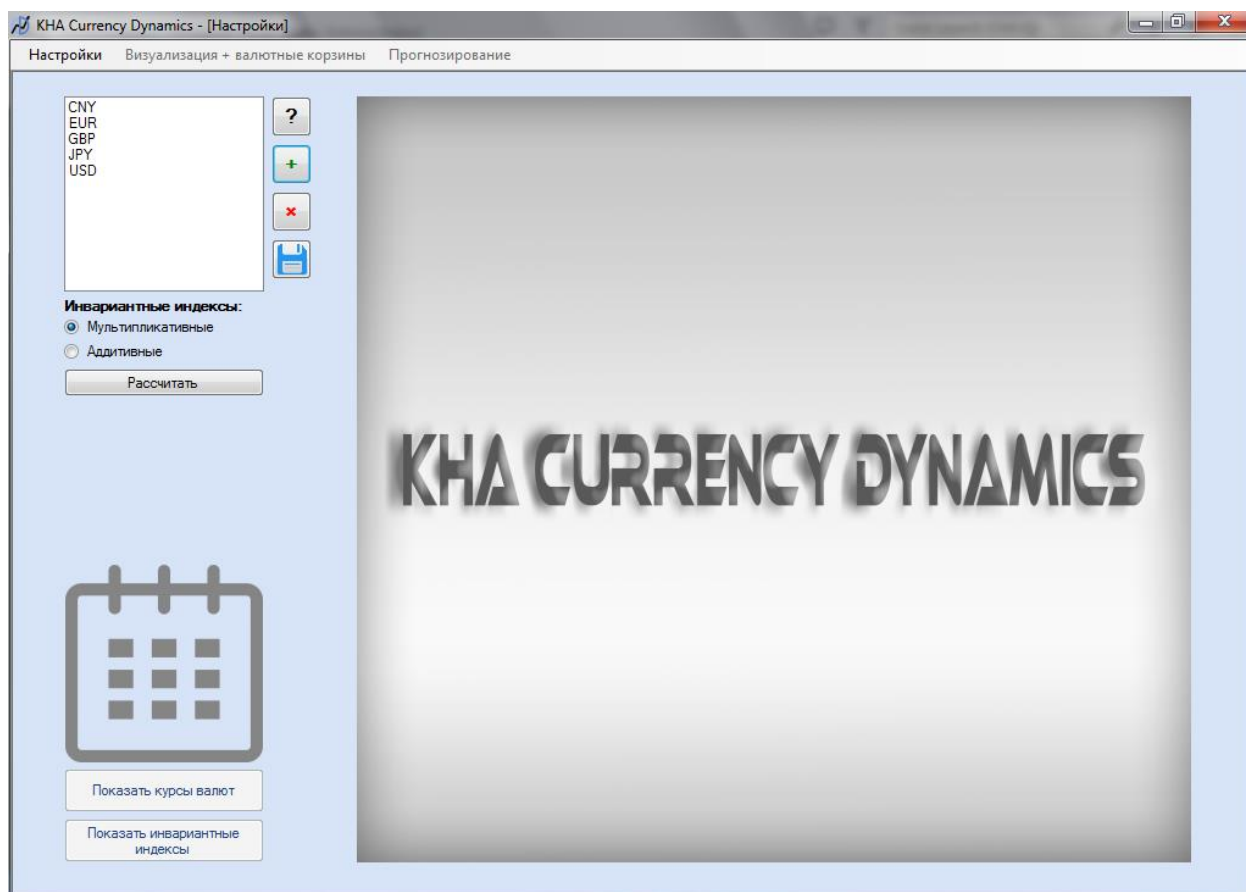


Рисунок 4.1.1. Стартовое окно.

Навигация по программе осуществляется с помощью главного меню, расположенного в верхней части экрана. Оно остается неизменным при переходе в разные разделы.

Прежде чем перейти к построению графиков, необходимо передать в программу входные данные и произвести их обработку. Для этого в верхней левой части окна нужно добавить в список файлы с расширением .dbf, каждый из которых содержит курс какой-то определенной валюты к рублю.

Данные файлы можно получить на почту, заполнив соответствующую форму на официальном сайте Центрального банка Российской Федерации. Временной промежуток у всех подключаемых файлов должен быть одинаковым.

Рядом со списком присутствуют следующие кнопки:



– Вызов краткой справки о требованиях, налагаемых на входные файлы.



– Добавление файла в программу. После нажатия откроется диалоговое окно, в котором можно выбрать подходящий файл, предварительно скачанный на жесткий диск или съемный носитель.



– Удаление выбранного файла из списка.



– Сохранение списка подключенных к текущему моменту файлов. При следующем запуске программы их не нужно будет добавлять снова.

После выбора всех необходимых входных файлов, следует произвести преобразования данных и расчёт инвариантных индексов валют согласно параграфам 3.1 и 3.2 нажатием одной кнопки «Рассчитать». В случае успешного подсчета появятся два новых графических элемента: календарь и таблица (рисунок 4.1.2).

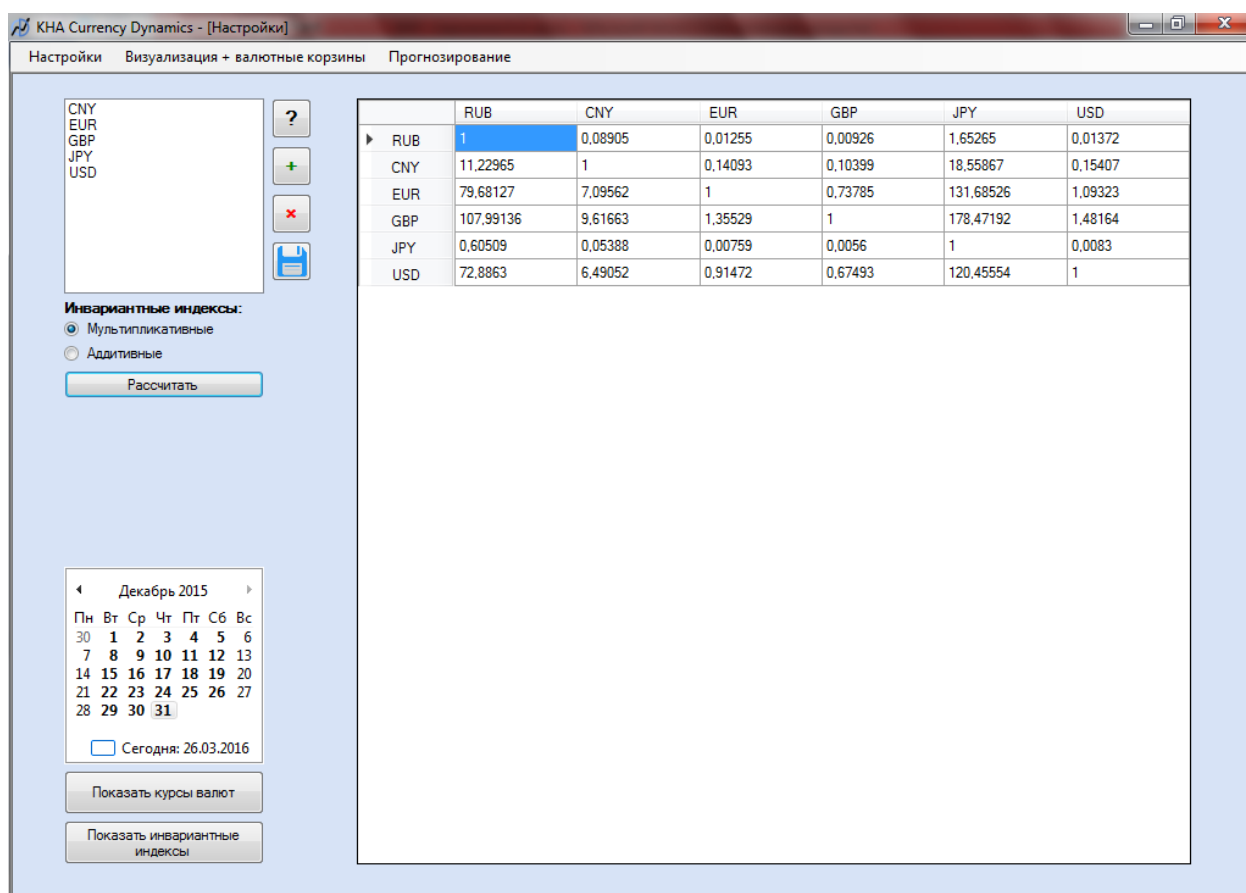


Рисунок 4.1.2. Окно настроек. Результат расчета индексов.

В правой части экрана можно увидеть матрицу курсов валют за какую-либо дату. Выбрать эту дату можно с помощью календаря. Также стали активными кнопки «Показать курсы валют» и «Показать инвариантные индексы». С их помощью можно переключаться между табличными представлениями курсов и индексов валют. Рисунок 4.1.3 демонстрирует вид окна программы с таблицей значений индексов.

KHA Currency Dynamics - [Настройки]

Настройки Визуализация + валютные корзины Прогнозирование

CNY
EUR
GBP
JPY
USD

?
+
x

Инвариантные индексы:

☒ Мультипликативные
☐ Аддитивные

Рассчитать

Показать курсы валют

Показать инвариантные индексы

| Date | RUB | CNY | EUR | GBP | JPY | USD |
|------------|---------|---------|---------|---------|---------|---------|
| 01.01.2015 | 1 | 1 | 1 | 1 | 1 | 1 |
| 13.01.2015 | 0,92009 | 1,02681 | 1,00081 | 0,99596 | 1,0347 | 1,02629 |
| 14.01.2015 | 0,89454 | 1,03277 | 1,00438 | 1,00252 | 1,04222 | 1,03145 |
| 15.01.2015 | 0,87834 | 1,03397 | 1,00157 | 1,00709 | 1,05761 | 1,03218 |
| 16.01.2015 | 0,89346 | 1,03293 | 0,99781 | 1,00624 | 1,04757 | 1,0302 |
| 17.01.2015 | 0,89028 | 1,03152 | 0,98822 | 1,00863 | 1,05873 | 1,03188 |
| 20.01.2015 | 0,89529 | 1,03192 | 0,98482 | 1,00632 | 1,05593 | 1,03433 |
| 21.01.2015 | 0,89659 | 1,03471 | 0,98774 | 1,00482 | 1,0485 | 1,03583 |
| 22.01.2015 | 0,8888 | 1,03513 | 0,98583 | 1,00996 | 1,05349 | 1,03625 |
| 23.01.2015 | 0,89088 | 1,03549 | 0,9874 | 1,00831 | 1,051 | 1,03597 |
| 24.01.2015 | 0,91996 | 1,03325 | 0,96758 | 0,99866 | 1,04967 | 1,03721 |
| 27.01.2015 | 0,89626 | 1,03725 | 0,96484 | 1,0084 | 1,05804 | 1,04495 |
| 28.01.2015 | 0,87031 | 1,0431 | 0,97121 | 1,01729 | 1,06275 | 1,04909 |
| 29.01.2015 | 0,875 | 1,03779 | 0,97645 | 1,0186 | 1,05971 | 1,04483 |
| 30.01.2015 | 0,85953 | 1,04356 | 0,97556 | 1,02036 | 1,06633 | 1,05034 |
| 31.01.2015 | 0,85723 | 1,04306 | 0,97978 | 1,01868 | 1,06677 | 1,05041 |
| 03.02.2015 | 0,8504 | 1,04431 | 0,98041 | 1,01798 | 1,07078 | 1,05367 |
| 04.02.2015 | 0,86966 | 1,03928 | 0,9772 | 1,01033 | 1,06972 | 1,0476 |
| 05.02.2015 | 0,89225 | 1,03142 | 0,97926 | 1,00986 | 1,05834 | 1,03823 |
| 06.02.2015 | 0,85857 | 1,03987 | 0,97902 | 1,0224 | 1,06804 | 1,04772 |
| 07.02.2015 | 0,88367 | 1,03133 | 0,97791 | 1,0214 | 1,05858 | 1,03775 |
| 10.02.2015 | 0,89069 | 1,03493 | 0,97172 | 1,02023 | 1,05028 | 1,04187 |
| 11.02.2015 | 0,89455 | 1,03506 | 0,97014 | 1,01953 | 1,04902 | 1,04091 |
| 12.02.2015 | 0,889 | 1,03811 | 0,97205 | 1,02342 | 1,04329 | 1,04402 |
| 13.02.2015 | 0,8879 | 1,03694 | 0,97231 | 1,02216 | 1,04737 | 1,04342 |
| 14.02.2015 | 0,89634 | 1,03161 | 0,97497 | 1,02464 | 1,04337 | 1,03756 |
| 17.02.2015 | 0,92479 | 1,02327 | 0,96779 | 1,01941 | 1,03966 | 1,03025 |
| 18.02.2015 | 0,93416 | 1,02452 | 0,96427 | 1,01774 | 1,04184 | 1,03278 |

Рисунок 4.1.3. Окно настроек. Табличное представление индексов валют.

В случае неудачной попытки расчётов (например, если пользователь попытался подсчитать индексы, не подгрузив ни одного входного файла с валютами), вместо таблицы со значениями в центральной части окна появится сообщение об ошибке.

Далее можно перейти к построению графиков. Соответствующий пункт главного меню после выполнения подсчетов становится активным.

4.2. Визуализация индексов валют и валютных корзин

В разделе визуализации в верхней части окна расположен график, а в нижней – инструменты для его построения (рисунок 4.2.1).

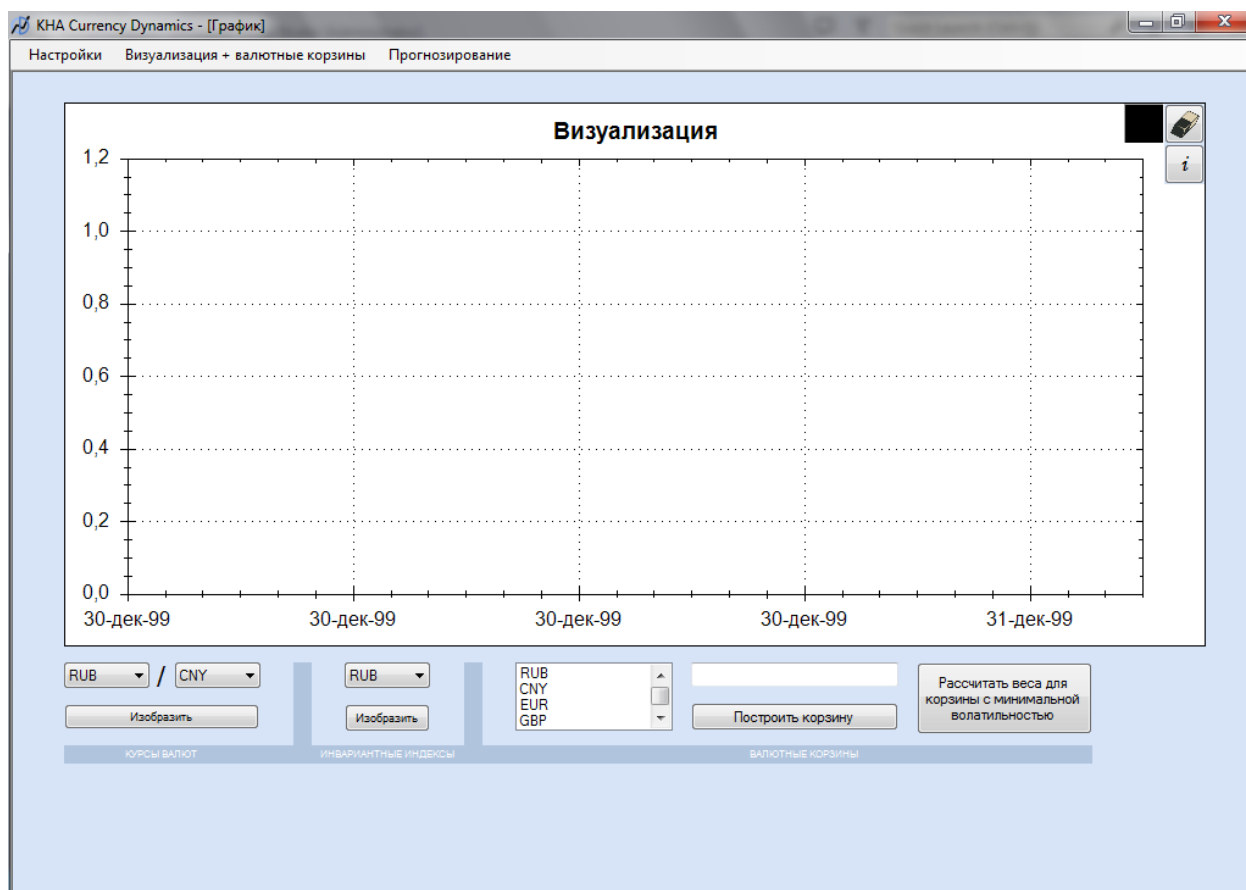


Рисунок 4.2.1. Окно визуализации.

Инструменты для построения графика разделены на категории: «курсы валют», «инвариантные индексы» и «валютные корзины». В первых двух категориях достаточно выбрать нужные валюты и нажать кнопку «Изобразить».

Для построения графика валютной корзины, необходимо из списка выбрать валюты, из которых будет состоять искомая корзина. Затем в текстовое поле ввести весовые коэффициенты для каждой валюты через пробел (самый левый коэффициент соответствует верхней валюте в списке, самый правый – нижней). Каждый из весов должен быть больше или равен

нулю и меньше или равен единице, а сумма весов должна равняться единице. После задействования кнопки «Построить корзину» её график появится в соответствующей области экрана.

Выбрав из списка валют нужные и нажав кнопку «Рассчитать веса для корзины с минимальной волатильностью», начнутся подсчеты, после выполнения которых, текстовое поле будет заполнено значениями весов, минимизирующих отклонение графика. Например, значения весов для наиболее стабильной корзины, построенной на основе данных за промежуток времени 01.01.2014 – 31.12.2015 и состоящей из российского рубля, китайского юаня, евро и доллара США, будут равны 0.172, 0.174, 0.289 и 0.364 соответственно. График такой корзины соответствует линии красного цвета на рисунке 4.2.2.

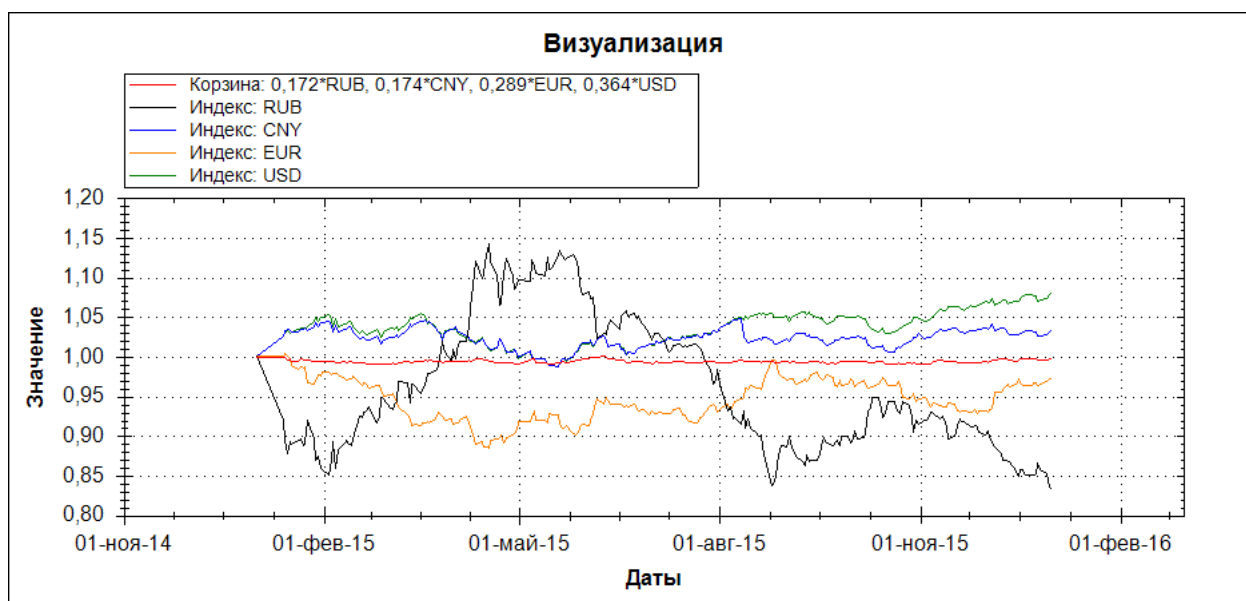


Рисунок 4.2.2. Пример построения стабильной валютной корзины.

Управлять видом графика можно с помощью контекстного меню, вызываемого нажатием правой кнопки мыши при её наведении на область, в которой расположен график. Присутствует возможность изменения масштаба, сохранения изображения на жесткий диск или в буфер обмена, а

также вывода на печать. В правом верхнем углу графика расположены кнопки очистки, выбора цвета линий и вызова краткой подсказки.

4.3. Прогнозирование

Для перехода к разделу прогнозирования можно воспользоваться соответствующим пунктом главного меню в верхней части окна.

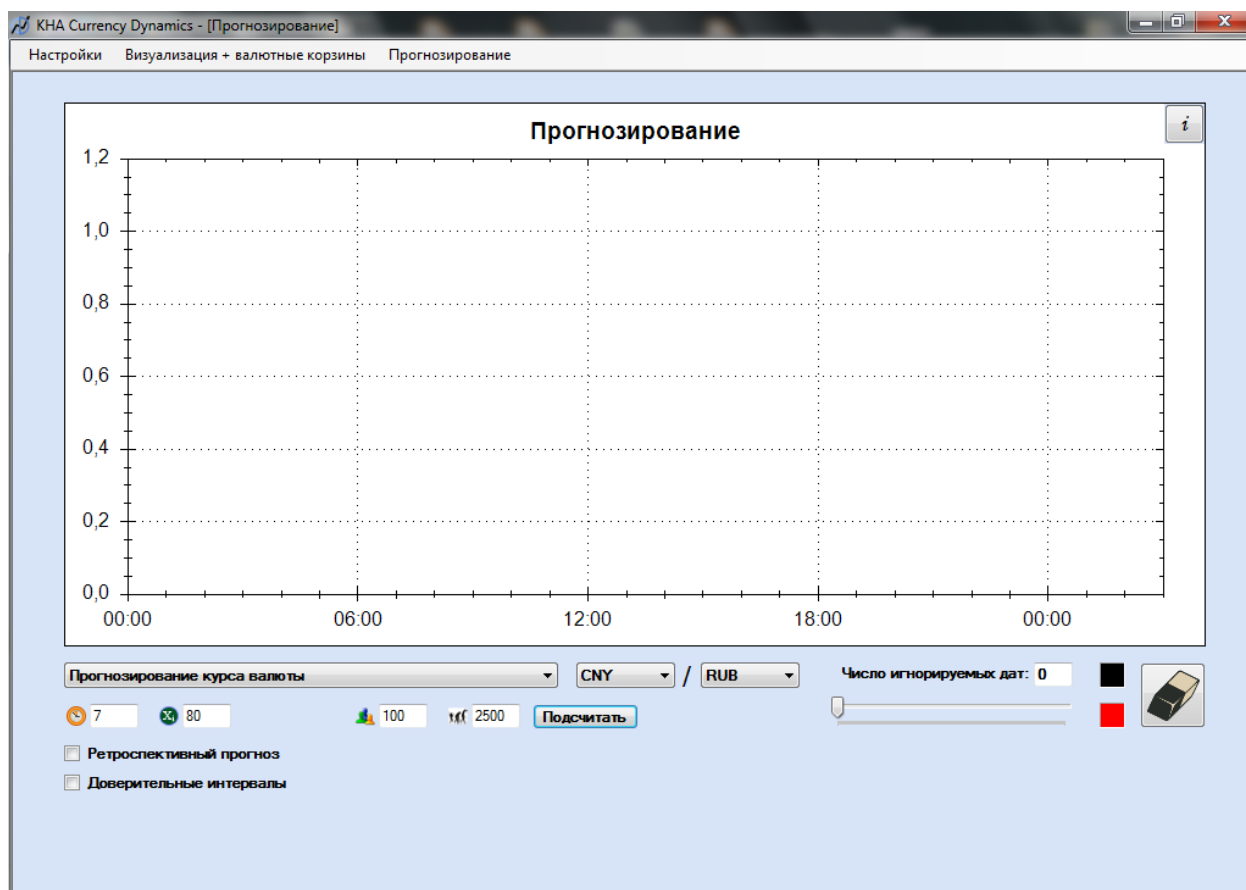


Рисунок 4.3.1. Раздел прогнозирования.

В верхней части данного раздела (рисунок 4.3.1) расположен график, обладающий всеми перечисленными в параграфе 4.2 свойствами. Единственным отличием является возможность выбрать разные цвета линий для фактических данных и для прогноза. Под графиком находится ряд выпадающих списков и текстовых полей, с помощью которых настраиваются параметры расчётов.

В левом выпадающем списке выбирается объект прогноза (рисунок 4.3.2).

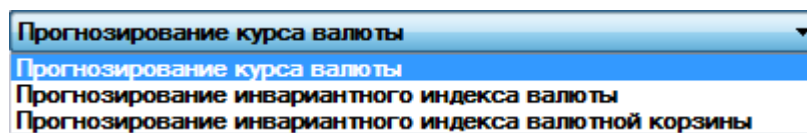


Рисунок 4.3.2. Виды прогнозов.

В первых двух случаях с помощью соседних выпадающих списков можно выбрать конкретные валюты, а в последнем вести работу с корзиной, построенной по инструкции из параграфа 4.2 (если таковая вообще построена).

Для более тонкой настройки предназначены поля со следующими иконками.



- Длина прогноза. Измеряется в днях. Рассматриваются только дни, в которые происходит изменение курсов валют (пять дней недели).



- Количество независимых переменных в процентном соотношении от тех данных, на которых происходит поиск оптимальных коэффициентов регрессии.



- Размер популяции в генетическом алгоритме минимизации ошибки аппроксимации.



- Количество шагов эволюции в генетическом алгоритме минимизации ошибки аппроксимации.

Все перечисленные поля для удобства пользователя заполнены значениями по умолчанию. Вдобавок к этому, при наведении курсора на любую иконку появляется всплывающая подсказка.

При установке галочки напротив пункта «ретроспективный прогноз» открывается календарь, на котором можно выбрать стартовую дату для данного прогноза. Аналогичным образом можно выбрать опцию построения доверительных интервалов.

При наличии большого объема исторических данных, подгруженных в программу, имеет смысл игнорировать часть из них для построения более

точного прогноза. Это возможно сделать с помощью соответствующего ползунка в правой части экрана.

В качестве иллюстрации приведем ретроспективный прогноз фунта стерлингов к рублю (GBP/RUB) на 7 дней с параметрами, представленными на рисунке 4.3.3.

The screenshot shows a software interface for currency forecasting. At the top, there is a dropdown menu labeled 'Прогнозирование курса валюты' and two currency selection dropdowns set to 'GBP' and 'RUB'. To the right, a slider is labeled 'Число игнорируемых дат: 138'. Below these are several input fields: a clock icon with '7', a percentage icon with '80', a person icon with '100', and a bar chart icon with '2500'. A 'Подсчитать' button is next to these fields. At the bottom, there are two checkboxes: 'Ретроспективный прогноз. Начиная с' (checked) and 'Доверительные интервалы' (unchecked). The date '01.12.2015' is entered in a field next to the first checkbox.

Рисунок 4.3.3. Параметры, использованные при прогнозе курса GBP/RUB.

В качестве результата получим график, изображенный на рисунке 4.3.4.



Рисунок 4.3.4. Прогноз курса GBP/RUB. Красная линия – прогноз, черная – реальные данные.

Заключение

В данной работе была рассмотрена экономическая модель, построены инвариантные индексы и обосновано их применение. Были формализованы такие понятия, как «валютная корзина» и «валютная корзина минимальной волатильности». Показана необходимость разработки программного обеспечения для анализа динамики валют, построения корзин и прогнозирования. Было создано ПО, обоснован выбор тех или иных инструментов, описаны использованные алгоритмы и продемонстрирована работа программы.

Разработанное программное обеспечение решает все поставленные задачи.

- Программа позволяет автоматизировать расчет курсов и индексов валют, имеет интерфейс табличного представления числовых значений данных показателей, а также инструменты для создания графика. Все это может использоваться для анализа динамики валют.
- У пользователя имеется возможность на основе входных данных строить различные валютные корзины и визуализировать их. Реализовано построение валютной корзины минимальной волатильности для разных наборов валют и разного количества валют в наборе.
- Разработан модуль прогнозирования. Пользователю предоставляется возможность изменения параметров прогноза. Данный модуль является инструментом для исследователя, позволяющим автоматизировать расчеты, экспериментировать с параметрами и визуализировать результаты. Точность прогноза в большой степени зависит от введенных пользователем настроек. Регрессионная модель, использованная в программе, ориентирована на построение

краткосрочного прогноза. В будущем возможно расширение данного модуля с целью добавления других методов прогнозирования.

- Программа позволяет обрабатывать входные данные, полученные при помощи интерфейса официального сайта Центрального банка Российской Федерации. Эти статистические данные являются надежными и корректными. При этом со стороны пользователя не требуется их дополнительная обработка перед использованием в разработанном ПО.
- Создан удобный графический интерфейс со всплывающими подсказками и краткими справками, что делает использование описываемого программного обеспечения удобным и минимизирует временные затраты на изучение возможностей программы.

Данная работа была представлена на XLVII Международной научной конференции аспирантов и студентов «Процессы управления и устойчивость», а также на XVI Всероссийском Симпозиуме по прикладной и промышленной математике. Тексты соответствующих докладов приняты к публикации в сборниках трудов конференций.

Список литературы

1. База данных по курсам валют | Банк России.
http://www.cbr.ru/currency_base/dynamics.aspx
2. Хитров Г. М., Хованов Н. В. Простая модель обмена: основные предположения и ближайшие следствия // Вестник Санкт-Петербургского университета. Серия 5: Экономика. 1992. № 4. С. 101–106.
3. Хитров Г. М., Хованов Н. В. Простая модель обмена: рандомизированные транзитивные матрицы коэффициентов обмена // Вестник Санкт-Петербургского университета. Серия 5: Экономика. 1994. № 1. С. 94–100.
4. Хитров Г. М., Хованов Н. В. Простая модель обмена: анализ динамики покупательной способности и курсовой стоимости валюты // Вестник Санкт-Петербургского университета. Серия 5: Экономика. 1995. № 3. С. 90–96.
5. Hovanov N. V., Kolari J. W., Sokolov M. V., Sutyurin S. F. Meta-money: theory and application under risk and instability // The Sixth International Scientific School “Modeling and Analysis of Safety and Risk in Complex systems” (MASR-2006). July 4–8, 2006. SPb., RAS, 2006. P. 240–247.
6. Hovanov N. V., Kolari J. W., Sokolov M. V. Synthetic U.S. Dollar as an “Anchor” for Chinese yuan (An application of currency stochastic indices theory) // International Scientific School “Modeling and Analysis of Safety and Risk in Complex systems” (MASR-2004). June 22–25, 2004. SPb., Russia. IPME, 2004. P. 219–225.
7. Hovanov N. V., Kolari J. W., Sokolov M. V. Computing currency invariant indices with an application to minimum variance currency baskets // Journal of Economic Dynamics & Control 28. 2004. P. 1481–1504.
8. Pontines V., Rajan R. S. The Asian Currency Unit (ACU): exploring alternative currency weights // Macroeconomics and Finance in Emerging Market Economies Vol. 1, No. 2, September 2008, P. 269–278.

9. Holland J. H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975. 211 pp.
10. Нечаев Ю. И., Дегтярев А. Б. Интеллектуальные системы: концепция и приложения. СПб.: Изд-во С.-Петербург. ун-та, 2011. 269 с.
11. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Язык программирования С#. Классика Computers Science. 4-е изд. СПб.: Питер, 2012. 784 с.
12. Финансовый словарь. http://dic.academic.ru/dic.nsf/fin_enc/12090
13. StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share. <http://gs.statcounter.com/#desktop-os-ww-monthly-201501-201601-bar>
14. Market share for browsers, operating systems and search engines | News. <http://www.netmarketshare.com/report.aspx?qprid=11&qpcustomb=0&qpsp=192&qpnp=13&qptimeframe=M&qpct=2>
15. ZedGraph. <https://sourceforge.net/projects/zedgraph/>
16. Li H., Love P. E. D., Ogunlana S. Genetic algorithm compared to nonlinear optimization for labour and equipment assignment // Building Research & Information, 1998. 26:6, P. 322–329.
17. Gallagher K., Sambridge M. Genetic algorithms: a powerful tool for large-scale nonlinear optimization problems // Computers & Geosciences Vol. 20, No. 7/8, 1994. P. 1229–1236.
18. А. П. Ротштейн “Интеллектуальные технологии идентификации”. http://matlab.exponenta.ru/fuzzylogic/book5/1_2.php